A novel computer vision based one class data driven modelling approach for person specific fall detection

Liyun Gong[@], Lu Zhang[*], Ming Zhu[*],Miao Yu[@], Ross Clifford[+], Carol Duff[+], Xujiong Ye[@] and Stefanos Kollias[@]

Abstract

In this paper, we propose a novel person specific fall detection system based on a monocular camera, which can be applied for assisting the independent living of an older adult living alone at home. A single camera covering the living area is used for video recordings of an elderly person's normal daily activities. From the recorded video data, the human silhouette regions in every frame are then extracted based on the codebook background subtraction technique. Low-dimensionality representative features of extracted silhouetted are then extracted by convolutional neural network based autoencoder (CNN-AE). Features obtained from the CNN-AE are applied to construct an one class support vector machine (OCSVM) model, which is a data driven model based on the video recordings and can be applied for fall detection. From the comprehensive experimental evaluations on different people in a real home environment, it is shown that the proposed fall detection system can successfully detect different types of falls (falls towards different orientations at different positions in a real home environment) with small false alarms.

Index Terms

healthcare, fall detection, data driven model,convolutional neural network autoencoder, one class support vector machine

@ Liyun Gong, Miao Yu, Xujong Ye and Stefanos Kollias are with School of Computer Science, University of Lincoln, UK (e-mail: {lgong, myu, xye, s.kollias}@lincoln.ac.uk).

+ Ross Clifford and Carol Duff are with School of Health and Social Care, University of Lincoln, UK (e-mail: {rclifford,cduff}@lincoln.ac.uk).

∗ Lu Zhang and Ming Zhu are with School of Computer Science and Technology, Shandong University of Technology, China (e-mail: {ming.zhu.1983}@gmail.com).

I. Introduction

There is an increasing number of aging populations across the globe. As shown in [1], the old-age dependency ratio (the number of people 65 and over relative to those between 15 and 64) is projected to 30% in the European Union and China, and to 20.2% in the United States. Among the elderly people over 65, a large number of them live independently. As shown in [2], more than three million people over 65 in UK live alone at home. The healthcare of elderly people living alone is becoming an increasingly important topic in the current aging society.

Falling is a dangerous activity, which seriously affects an elderly person's health and commonly happen among the old people community. According to the Public Health England [3], the number of people aged 65 or over that are expected to fall once a year is estimated to be 30% and for those aged 80 or over this percentage grows to 50%. Falling can lead to serious physiological and psychological damages to an elderly person's health conditions. Falling is among the three top-most common known causes of Traumatic Brain Injury (TBI) in the United States according to [4]. An estimated 10% of all falls in seniors cause major injuries, including intracranial injuries (ICIs) and fractures as in [5]. Moreover, according to the report of the World Health Organisation (WHO) [6], falling is the second leading cause of death by accidental or unintentional injury worldwide. Even when the injuries caused by falling are not so serious, fallers often struggle to get up unaided, which leads to a long-lie where the faller remains trapped on the floor for an extended period of time. Long-lies can lead to dehydration, pressure sores, pneumonia, hypothermia and even death [7].

For providing qualified healthcare services for the elderly people living independently, it is important to ameliorate the negative effects of falling thus an effective fall detection system is indispensable. When an elderly person falls, such a system can detect this dangerous activity as it happens and related alarming messages can be sent to nurseries/family members who then provide timely healthcare assistances to ameliorate negative consequences caused by falling. During recent decades, there have been numerous proposed fall detection systems, which can be deployed domestically for automatically detecting falls of elderly adults living alone at home. Wearable sensors based methodologies have been exploited for the fall detection, examples include [8], [9]. Signals are collected from wearable triaxial accelerometer. Corresponding features are extracted from signals and fed into Cascade-AdaBoost-SVM [8] and Hidden Markov Model (HMM) [9] for classifications of falls and normal activities. In [10]–[12], acoustic sensors based fall detection approach is adopted. The acoustic signals are collected from both normal activities and falls. Mel-frequency cepstral coefficients (MFCCs) are extracted from raw collected acoustic signals

for various of classifiers (such as k-nearest neighbor classifier (k-NN), support vector machine (SVM), least squares method (LSM), and artificial neural network (ANN) as reported in [10]) have been evaluated for classifying fall/non-fall activities. The mobile phone based solutions for fall detection are proposed in [13], [14], in which motion information, such as speed, acceleration and angle are extracted by the built-in accelerometer in a mobile phone to determine whether a fall occurs or not. Alwan et al. [15] have developed a passive system to the occupant, which is floor vibration-based fall detector system. By using spring and mass arrangement, piezoelectric sensors can be coupled to the surface of the floor. By analyzing patterns of vibration signals collected from piezoelectric sensors, this system differentiates the fall event from other daily activities. And in [16], [17], Doppler radar is used for detecting falls based on analyzing of Doppler radar motion signature.

However, the aforementioned fall detection methodologies have unavoidable limitations. For the wearable sensors/mobile phone based methodologies for fall detection, an elderly person has to wear a triaxial accelerometer or bring a mobile phone all the time during his daily life and the batteries of these sensors need to be recharged frequently, which incurs inconveniences for daily living. For the acoustic sensors based methodologies, their performance can be easily deteriorated by different types of noises (such as TV noises, noises from outdoor traffic) existing at home. The floor vibration sensors based methods have a limitation with respect to the detection range; besides, piezoelectric sensors can not effectively acquire vibration signals for all sorts of floor materials. And the received Doppler radar motion signature may be affected by other moving objects in a real home environment, thus generating miss detections or false alarms.

Considering these limitations, currently computer vision based fall detection system becomes the mainstream one, which exploits the ordinary RGB camera or Kinect together with modern computer vision/machine learning techniques for detecting falling events. Compared with other fall detection approaches, such as wearable sensors based ([8], [9]), mobile phone based methods ([13]), acoustic sensors based ones ([10]–[12]) and floor vibration based method ([15]), the computer vision based methods have their unique advantages. Mostly they are non-intrusive (an elderly person needs not wear some special equipment such as an accelerometer or bring a mobile phone all the time); besides, they are not easily affected by noises, such as TV noises in the surrounding environment.

The computer vision based fall detection methods can be divided into two main categories: threshold based approaches and machine learning based approaches:

i. Threshold based approaches: For the threshold based approaches, extracted features from camera recordings and compared with certain thresholds to determine whether a fall occurs or not. Calibrated

cameras are used to reconstruct the three-dimensional shape of people in [18], and fall events were detected by analyzing the volume distribution along the vertical axis. When the major part of this distribution was abnormally near the floor during a predefined period of time, an alarm indicating a fall was triggered while a graphic processing unit (GPU) was applied for accelerating the computations. In [19], the 3D head information (position and velocity) is tracked by the particle filtering approach by a single calibrated camera; while the 3D vertical velocity or height obtained from the tracking results are compared with preset thresholds to detect falls. Instead of applying a fixed camera, a wearable camera based method is applied in [20], which is capable of detecting falls without limiting to a particular area. Both the gradient local binary patterns (GLBP) and edge orientation (EO) features are calculated. The product of dissimilarity values of GLBP and EO between two consecutive frames is calculated to compare a threshold to determine whether a fall happens or not. The main limitation of the threshold based approaches is that it is impossible to find suitable thresholds to detect the variety of falls occurred in reality (e.g., falls with large/small movement amplitudes, falls in different orientations,etc.).

ii. Machine learning based approaches:

Machine learning algorithms have also been incorporated into the computer vision based fall detection methods. In [21], a person's three-dimensional orientation information from multiple cameras is extracted. From extracted orientation information from a short video sequence, an improved version of HMM– layered hidden Markov model (LHMM) was used to distinguish normal walking and falls. In [22], different types of activities (walking, sitting, sweeping, falling, etc.) are simulated by volunteers and recorded by video camears. With respect to video recordings, the silhouette areas obtained from a number of consecutive frames extracted from running Gaussian average background subtraction are taken as features, which are fed into a multi-class support vector machine (SVM) for classifying different types of normal activities and falls. Instead of visual images, a Kinect sensor is used in [23] which applies the recorded depth images for fall detection. Firstly, the 3-D human body is reconstructed and tracked by the Kinect from the background subtraction results of recorded depth imageries. Related 3-D features, such as minimum vertical velocity (MVV), maximum vertical acceleration (MVA) of the reconstructed 3-D body are then extracted to classify whether a fall happens or not based on the decision tree algorithm. With the development of deep learning and its high performance in different traditional computer vision tasks (object detection/classification), it has also recently received attentions in the fall detection community. In [24], a fall detection system is devised based on a convolutional neural network (CNN) classifier for postures classification. Falling is reported when it is detected that a person lies on the ground for a certain period. It is the first work which introduces the CNN for fall detection and from the results, it

is shown that the CNN based approach outperforms the traditional SVM based ones for daily postures classification. Further to [24], a three-dimensional convolutional neural network (3-D CNN) based method for fall detection is developed in [25], which uses video kinematic data to train a deep neural network for automatic features extraction and classifying different types of activities including falling. Different machine learning approaches are applied/compared in [26] for fall detection based on both real-world and simulated data.

With respect to the current machine learning approaches, different types of classifiers (such as traditional ones in [21]–[23] and deep learning based ones in [24], [25]) are constructed for fall detection, based on particular recorded video training datasets containing both simulated normal activities and abnormal ones such as falling by experimental volunteers. However, the layout of an experimental site for recording such training datasets may be different from that of a real domestic environment. The background in a particular real home environment may be more complicated with more furniture, which can cause human body occlusions in the view of a camera. Besides, the activity types/characterisitcs in training datasets may be different from those of a specific elderly person living alone at home. All these can deteriorate the performance of corresponding constructed classifiers for classifying normal/abnormal activities for a specific person at home.

In this work, we propose a new one class data driven modelling based computer vision fall detection method, which constructs a model based on recorded video data for *person specific* fall detection. In specific, the daily activities of a specific elderly person will be recorded by normal RGB camera, codebook background subtraction method is applied to extract the human body silhouettes from video recordings. Instead of hand-crafted features needing much empirically parameters setting, convolutional neural network based auto-encoder is applied to automatically extract representative features from silhouettes. Based on extracted features, an one class support vector machine (OCSVM) model is constructed for fall detection. Compared with its counterparts, the advantages of the proposed approaches include the followings:

i). The proposed method provides a *person-specific* solution. Both the features and OCSVM model used for fall detection, are obtained based on the video recording of that specific person's daily activities at his/her home instead of from other datasets. In this way, the constructed fall detection system can be more suitable to distinguish normal activities and falls for that specific person.

ii). We proposed a novel CNN-AE based approach for feature extraction, which can extract more representative features to achieve more accurate fall detection performance compared with other features (as shown in the experimental study).

iii). Compared with other all detection approaches like [21], [23], [24] and [25] which need data from

both normal and falling activities for constructing a machine learning model to detect falls, the proposed approach only relies on normal activities data without needing the fall activities data, which reduces the data storage costs and computational costs for model construction as well as the laborious process of collection/simulation of falling data for machine learning model construction.

The works in [27]–[29] are similar to ours, which also only rely on normal activities data for fall detection. However, these works rely on the Gaussian mixture model (GMM) which imposes an assumption that modelled normal activities data has to follow a mixture of Gaussian distribution. Instead of GMM, one class support vector machine (OCSVM) model is exploited in our work. OCSVM model can flexibly model any data distributions not only limited to mixture of Gaussian one and from the experimental study, it is shown that our proposed OCSVM model achieves better performance than GMM model for fall detection. The paper is structured as follows: A general overview of the proposed method is shown in Section II. Section III shows details of the proposed system for fall detection ,including the background subtraction, features extraction and normal model construction. Section IV shows the related experimental evaluations of the proposed fall detection method. Conclusions and future works are finally given in Section V.

II. Methodology

The flow chart of the proposed system is presented in Fig. 1. Video streams of daily activities are recoded by a camera. For each frame within recorded video streams, codebook background subtraction is firstly applied to extract the human silhouette regions, which are the areas of interests for our further analysing. We exploit the convolutional neural network based auto-encoder (CNN-AE), which is one type of the unsupervised deep neural network, for extracting representative features from extracted silhouette regions. The extracted features are applied for training a one class support vector machine (OCSVM). Then for a new testing incoming video sequence, the human silhouettes in the sequence are extracted through background subtraction. Corresponding features from silhouettes are extracted by CNN-AE and fed into OCSVM, for testing whether these silhouettes represent normal activities or not.

The details of every block in Fig. 1 will be discussed in the next few subsections.

A. Background Subtraction

Background subtraction is a common approach for discriminating objects of interests from the background in visual surveillance. In our fall detection system, we use codebook method [30] for extracting human silhouettes from video sequences because of the following merits as proposed in [30]: i). no

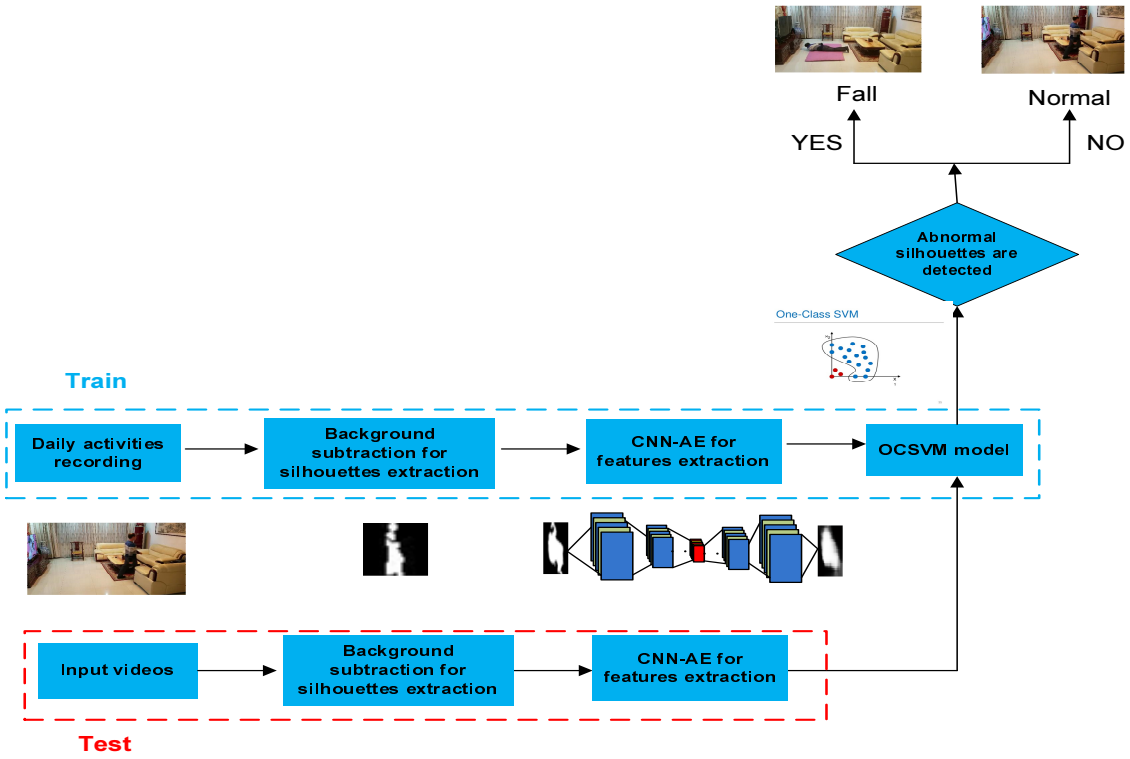

Fig. 1. The flow chart of the proposed fall detection system.

parametric assumption on the codebook model ii). resistance to artifacts of acquisition, digitization and compression; iii). capability of coping with illumination changes; iv). unconstrained training that allows moving foreground objects in the scene during the initial training period. And compared with deep neural network based human silhouettes extraction approaches like Mask-RCNN in [31], codebook background subtraction methods is much more light-weighted which does not need to train/store a complex deep neural network through dedicated hardware resources (such as GPU), which facilitates it to be used in the real-life applications.

The codebook method is available for both colour and gray-scale images. A codebook is initially constructed, which contains codewords obtained from a number of training images representing background information for every image pixel. For each codeword denoted by **c**, it consists of an RGB vector $\mathbf{v} = (R, G, B)$ and a 6-tuple $\mathbf{aux} = (\hat{I}, \check{I}, f, \lambda, p, q)$. The meanings of the six parameters in **aux** are described in TABLE I:

The details of the training procedure are given in [30] and the obtained codewords in the codebook are then used for background subtraction purpose. For an incoming colour frame **f**, its pixel $\mathbf{f}(x, y) =$

TABLE I
DEFINITIONS OF THE SIX PARAMETERS IN **AUX**

$\hat{I}$	**Maximum intensity that has been represented by the codeword.**
$\check{I}$	**Minimum intensity that has been represented by the codeword.**
f	**Number of times that the codeword has been used.**
λ	**Maximum negative runtime length (MNRL) in number of frames.**
p	**The first frame in which this codeword was used.**
q	**The last frame in which this codeword was used.**

$(R(x,y), G(x,y), B(x,y))$ (a 3-dimensional vector) is determined as a foreground or background pixel by comparing $\mathbf{f}(x,y)$ with corresponding codewords of that pixel. If $\mathbf{f}(x,y)$ is not matched with any codeword, then it is a foreground pixel. For a particular codeword $\mathbf{c}$, we say the codeword $\mathbf{c}$ matches $\mathbf{f}(x,y)$ if the following two conditions are met.

$$\begin{aligned} colordist(\mathbf{f}(x,y), \mathbf{c}) &\leq \varepsilon \\ brightness(I, \langle \hat{I}, \check{I} \rangle) &= true \end{aligned} \tag{1}$$

where ε is threshold value set manually, I represents the L2-norm of $\mathbf{f}(x,y)$, $\hat{I}$ and $\check{I}$ are the first two parameters of the 6-tuple **aux** vector of the codeword $\mathbf{c}$.

The $colordist(\mathbf{f}(x,y), \mathbf{c})$ measures the chromatic difference between two colour vectors, which can be calculated by:

$$colordist(\mathbf{f}(x,y), \mathbf{c}) = \sqrt{\| \mathbf{f}(x,y) \|^2 - \frac{\langle \mathbf{f}(x,y), \mathbf{v} \rangle}{\| \mathbf{v} \|^2}} \tag{2}$$

where $\mathbf{v}$ represents the RGB vector $\mathbf{v} = (R, G, B)$ of codeword $\mathbf{c}$, and $\| \cdot \|$ and $\langle \cdot \rangle$ denote respectively the L2-norm and dot product operations.

The $brightness(I, \langle \hat{I}, \check{I} \rangle)$ is defined as:

$$brightness(I, \langle \hat{I}, \check{I} \rangle) = \begin{cases} true & if \quad I_{low} \leq \| \mathbf{f}(x,y) \| \leq I_{hi} \\ false & \text{otherwise} \end{cases} \tag{3}$$

where $I_{low} = \alpha \hat{I}$ and $I_{hi} = min\{\beta \hat{I}, \frac{\check{I}}{\alpha}\}$. In our experiment, α and β are fixed to be 0.5 and 2 for background subtraction, which have been found empirically to be suitable values.

The obtained raw background subtraction results generally contain many noise artifacts, which include small “salt and pepper” noises and large noises caused by movement of furniture. In order to remove such noises, some post-processing ([32]) can be applied to improve the background subtraction results. Background subtraction examples are shown in Fig. 2. When a human object appears in the camera view, its silhouette region is extracted from the original video recording.

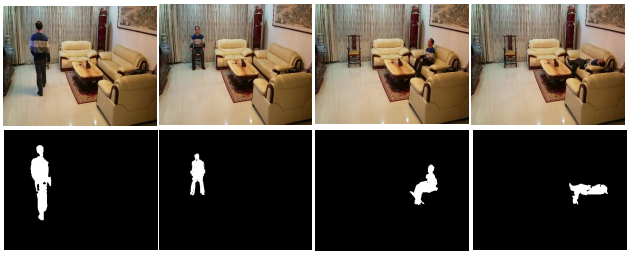

Fig. 2. Background subtraction results for representative frames of daily activities.

B. CNN-AE for features extraction

The human silhouette regions from the obtained background subtraction results are extracted and normalized to 30*30 image patches. Based on obtained image patches, representative features are then extracted by the convolutional neural network based autoencoder (CNN-AE), which is one type of autoencoders for data reconstruction/representation. Compared with other types of autoencoders, such as the standard autoencoder (AE) and stacked autoencoder (SAE), the CNN-AE have much fewer parameters to be tuned which makes it much easier to be trained based on a relatively small training dataset. Besides, by exploiting features obtained from CNN-AE, better fall detection performance can be obtained compared with the exploitation of other features (such as original image patches, ellipse features, principle component analysis (PCA) features, etc.). Related comparison results are shown in the experimental section.

The architecture of CNN-AE is presented in Fig. 3. It includes an encoding part which encodes the original input image into low-dimensionality representations in a latent space as well as a decoding part which reconstructs the original image. Each part contains multiple layers associated with different operations of convolution, downsampling and upsampling.

For the convolutional layers, convolutional kernels are exploited to obtain a set of representative feature maps. In specific, the value at the position (i,j) of the k-th feature map in the l-th convolutional layer

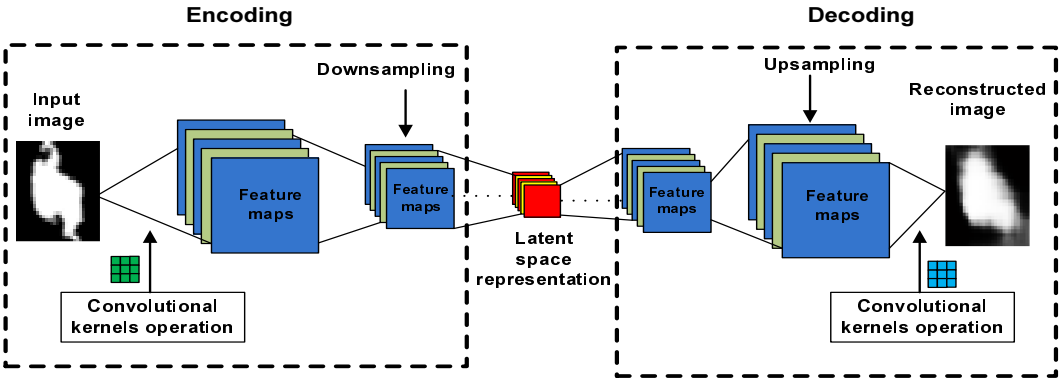

Fig. 3. Architecture sketch of CNN-AE

(denoted as $a(i,j)_k^l$) can be calculated by convolutional kernels and feature map values of the last layer as:

$$\begin{aligned} net(i,j)_k^l &= \sum_{d=0}^{N-1}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} w_{d,m,n}^{k,l} a(i+m, j+n)_d^{l-1} \\ &+ w_b^{k,l} \\ a(i,j)_k^l &= f(net(i,j)_k^l) \end{aligned} \tag{4}$$

where $net(i,j)_k^l$ represents the net input corresponding to the position (i,j) of the k-th feature map in the layer l as in [33], $f(\cdot)$ represents an activation function (e.g., sigmoid, Relu, etc.). $w_{d,m,n}^{k,l}$ represents the element at position (d,m,n) of the k-th 3-D kernel at layer l denoted as $W^{k,l}$ and $w_b^{k,l}$ is the biased value. In the downsampling layers, feature maps are downsampled by a factor remove redundancies. The upsampling layers are applied to upsample feature maps for reconstructing the input data.

For training the CNN-AE, different training algorithms, such as SGD, Adam or RMSProp [34] can be exploited based on the definition of a loss function. In our work, the loss function is defined as the root-mean-square-error between the input/output image patches.

C. OCSVM

The obtained CNN-AE features from silhouettes extracted from recordings of a person's normal daily activities/postures (e.g., waling and sitting), are applied to construct an one class support vector machine(OCSVM) model [35], which is a data driven modelling approach for data description. Compared with traditional data driven modelling approaches such as the single Gaussian model and Gaussian mixture model in [36], OCSVM model is more general which can flexibly describe/model a set of data with an

arbitrary distribution. The basic idea behind the OCSVM is that given a training dataset $X = [\mathbf{x}_1, ..., \mathbf{x}_N]$ drawn from an underlying probability distribution P, the estimted OCSVM based on the training dataset is a function $f(\mathbf{x})$ shown as:

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) - \rho \tag{5}$$

where $\mathbf{x}$ represents data sample and $\Phi(\cdot)$ is a mapping function which maps the data into high-dimensionality space. The function $f(\mathbf{x})$ is larger than a threshold if its input $\mathbf{x}$ is drawn from P as those samples in the dataset X. Otherwise, it is less than the threshold. In this way, the OCSVM model trained based on a particular training dataset, can be used to determine whether a new input data belongs to the same class as the samples in the training dataset.

The parameters in (5) which need to be estimated include $\mathbf{w}$ and ρ. In order to obtain the parameters $\mathbf{w}$ and ρ, the following quadratic problem needs to be solved:

$$\begin{aligned} \min_{\mathbf{w},\xi,\rho} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu N}\sum_i \xi_i - \rho \\ \text{subject to} \quad & (w \cdot \Phi(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0 \end{aligned} \tag{6}$$

where $\nu \in (0, 1]$ and the nonzero slack variables $\xi = [\xi_1, ..., \xi_N]$ are introduced to compensate for possible outliers in the training dataset as in [35].

As mentioned in [35], a dual form of problem (6) can be obtained by introducing Lagrangian multipliers as:

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu N}, \quad \sum_i \alpha_i = 1 \end{aligned} \tag{7}$$

where α_i represents a Lagrangian multiplier. According to the Karush-Kuhn-Tucker (KKT) condition, the relationship between the optimal function parameters (denoted as $\mathbf{w}^*$ and ρ^*) and optimal solution of α^* of (15) can be obtained as:

$$\mathbf{w}^* = \textstyle\sum_i \alpha_i^* \Phi(\mathbf{x}_i) \tag{8}$$

$$\rho^* = (\mathbf{w}^* \cdot \Phi(\mathbf{x}_i')) \tag{9}$$

where $\mathbf{x}_i'$ represents a data sample whose corresponding α_i^* follows $0 < \alpha_i^* < \frac{1}{\nu N}$

By substituting (8) into (5), we can finally obtain the function form as:

$$f(\mathbf{x}) = \sum_j \alpha_j^* \Phi(\mathbf{x})\Phi(\mathbf{x}_j) - \sum_j \alpha_j^* \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_i')) \tag{10}$$

Usually there is a no explicit form for representing $\Phi(\mathbf{x})$ and a kernel function is usually introduced as: $k(x, y) = \Phi(x)\Phi(y)$ [35]. By exploiting the kernel function, the above equation can be written as:

$$f(\mathbf{x}) = \sum_j \alpha_j^* k(\mathbf{x}, \mathbf{x}_j) - \sum_j \alpha_j^* k(\mathbf{x}_j, \mathbf{x}_i')) \tag{11}$$

Different kernels can be chosen. In this work, we have tested both the linear kernel: $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$ and the Gaussian kernel–$k(\mathbf{x}, \mathbf{y}) = exp(-\gamma\|\mathbf{x} - \mathbf{y}\|)$ with different kennel parameter γ. Related comparison results are provided in the experimental section.

1) Fall detection by the OCSVM: The trained OCSVM model based on CNN-AE features extracted from silhouettes representing normal activities/postures, can be applied to detect whether a fall happens or not based on the principle that when a person falls, his/her postures will be different from the ones associated with his/her normal daily activities (such as walking or sitting). In specific, given a video sequence, for every frame the silhouette is firstly extracted by the background subtraction method and CNN-AE is applied to extract features (denoted as $\mathbf{x}$), then we decide whether the silhouette in that frame corresponds to a normal posture or not based on the following criteria:

$$f(\mathbf{x}) \geq th \rightarrow normal, \qquad f(\mathbf{x}) < th \rightarrow abnormal \tag{12}$$

where $\mathbf{x}$ represents the extracted feature from a video frame and th is a preset threshold. If consecutive abnormal postures are detected, or the number of frames containing abnormal posture during the video sequence exceeds a preset threshold (indicating a person's status is abnormal, such as lying on the ground for a time period), then a falling activity is confirmed.

III. Experimental studies

In this section, we present the related experimental evaluations for the proposed fall detection system.

A. Experimental settings

The experiment was performed in a real-home environment where the elderly people live, as shown in Fig. 4. We used a normal personal laptop with a configuration of Intel Core Two 2.10GHz CPU with a 1.00GB memory for data processing. A USB camera was connected to the laptop for recording the

video streams, which was placed on the wall of the room close to the ceiling to cover the full view of the home environment. The video sequence was recorded at a frame rate of 15 frames/s.

Fig. 4. The home environment for the experimental studies

Due to the fact that the proposed fall detection methodology is a person-specific solution, one experimental volunteer is invited to participate in the experiments for the fall detection system evaluation. We interviewed a 75 years old, healthy old person and the frequency of representative activities during one week is summarized in Table I. According to this table,the experimental volunteer performs 38 normal activities(including 16 walking, 6 standing, 8 sitting and 8 lying), which are recorded by the USB camera as the training dataset. Totally, the training dataset contains 3,171 frames. Moreover, the experimental volunteer also performs additional 25 normal activities and 14 falls (at different orientations and positions in the room, even with occlusions), which are recorded as the testing dataset.

TABLE II
SUMMARY OF THE FREWUENCY OF REPRESENTATIVE ACTIVITIES OF AN ELDERLY PERSON DURING ONE WEEK

Activities	Descriptions	Frequency
Walking	The elderly person walks to move between different places of the room	16
Standing	The elderly person stands almost still at a particular position	6
Sitting	The elderly person sits to have a rest (such as watching TV)	8
Lying	The elderly person lies on the sofa for a nap or watching TV	8

B. Silhouette and features extraction

From original video recordings, related silhouettes are extracted by the codebook (CB) background subtraction method. Fig. 5 shows the background subtraction results from different types of activities. We also compared the performance of the codebook background subtraction methods with other two popular

background subtraction methods, such as threshold based background subtraction method and Gaussian mixture model (GMM) based one [37]. The background subtraction results for these three methods for some representative frames in a video sequence on a video sequence are shown in Fig. 6, intuitively, we can see the results achieved by the codebook method is most close to the groundtruth ones.

Moreover, we calculate the precision and recall for quantitatively measuring performance of three background subtraction methods. The definition of precision and recall related to background subtraction are defined as follows:

$$Precision = \frac{No.\ of\ correctly\ detected\ foreground\ pixels}{No.\ of\ totally\ detected\ foreground\ pixels} \tag{13}$$

$$Recall = \frac{No.\ of\ correctly\ detected\ foreground\ pixels}{No.\ of\ groundtruth\ foreground\ pixels} \tag{14}$$

ideally, all the two metrics should be 1. Fig. 7 and Fig. 8 show the comparisons of precision and recall values of background subtraction results for every frame of a video clip, by threshold-based, GMM and CB background subtraction methods. We can see the CB method can always achieve the highest values precision and recall values indicating the best performance.

As mentioned in the previous section, the silhouettes regions are extracted and normalized into 30×30 image patches, which are fed into CNN-AE for feature extractions. The architecture of the CNN-AE applied for features extraction contains seven layers, which are summarized in the Table III. For an image patch, the output of the sixth layer is taken and reshaped to be a 64-dimensionality feature vector.

TABLE III
ARCHITECTURE OF THE CNN-AE APPLIED FOR FEATURES EXTRACTION

Layers	Descriptions
1	Convolutional layer, with 32, 3-by-3 convolution kernels
2	Pooling layer, with the 2*2 maxpooling kernel
3	Convolutional layer, with 16, 3-by-3 convolution kernels
4	Pooling layer, with the 2*2 maxpooling kernel
5	Convolutional layer, with 4, 3-by-3 convolution kernels
6	Pooling layer, with the 2*2 maxpooling kernel
7	Upsampling layer, with a upsampling factor of 2
8	Convolutional layer, with 4, 3-by-3 convolution kernels
9	Upsampling layer, with a upsampling factor of 2
10	Convolutional layer, with 16, 3-by-3 convolution kernels
11	Upsampling layer, with a upsampling factor of 2
12	Convolutional layer, with 32, 3-by-3 convolution kernels
13	Convolutional layer, with 1, 3-by-3 convolution kernels

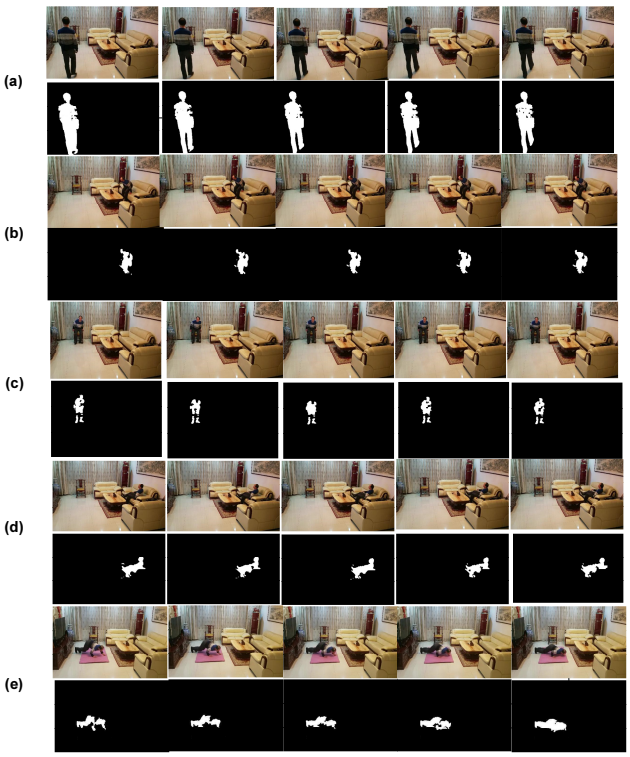

Fig. 5. The silhouettes extracted for different daily activities.

From the layers 1-12, the activation function is “ReLu” and the activation function for the last layer is “Sigmoid”. The training of the CNN-AE is based on the RMSProp algorithm implemented in the Keras [38], which is an open-source neural-network library written in Python.

Based on input image patches, CNN-AE can extract features and reconstruct image patches. The reconstructed image patches corresponding different postures (stand, sit , lie) by CNN-AE are shown in Fig. 9, which show that the image patches can be successfully reconstructed by CNN-AE. Fig. 10 shows the visualized feature vectors (L1 to L9) extracted from corresponding nine silhouettes (I1 to I9). From this figure, we can observe for silhouettes belonging to the same activity, the extracted feature patterns by the CNN-AE are similar. For silhouettes belonging to different activity, the extracted feature patterns by the CNN-AE are different. Intuitively, the CNN-AE features can be applied for distinguishing silhouettes of different activities.

We have compared the average reconstructed cross-entropy error of 3,171 human silhouettes extracted from video frames in the training dataset between three types of autoencoders,which include the original version of autoencoder (AE) and stacked autoencoder (SAE) and the CNN-AE adopted in our work.

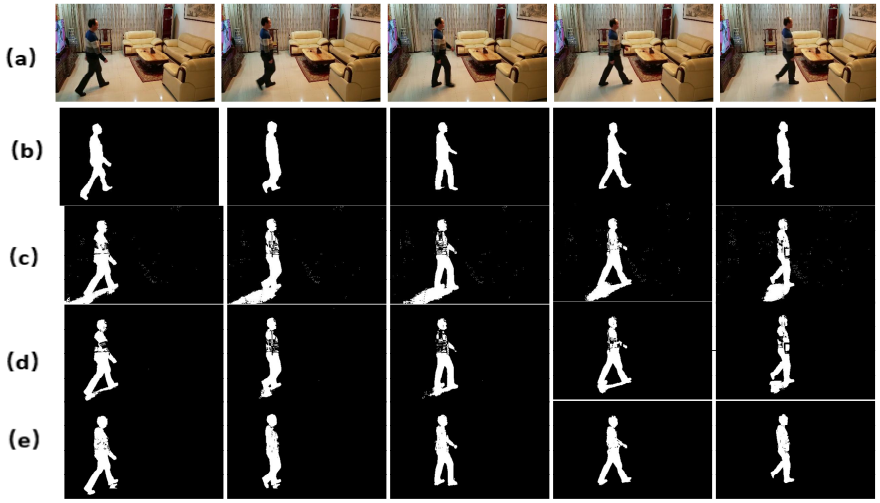

Fig. 6. Comparisons of silhouette extractions by different background subtraction methods. (a). original video frames (b). ground truth silhouettes (c). silhouettes extracted by threshold-based background subtraction (d). silhouettes extracted by GMM background subtraction (e). silhouettes extracted by CB background subtraction

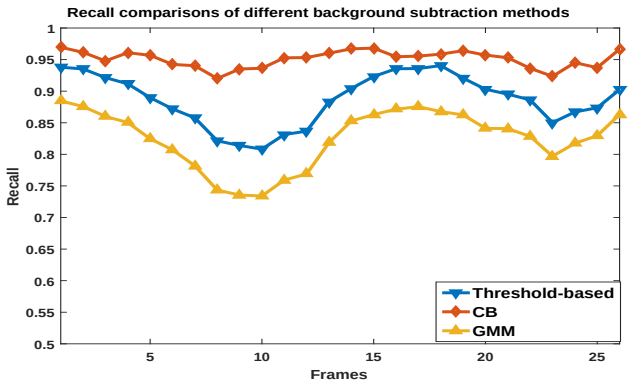

Fig. 7. Recall comparisons of different background subtraction algorithms for different frames in a video sequence

Same as the CNN-AE, AE and SAE used for comparison encode the original 30*30 image patches to 64-d feature vector. The comparison results are presented in Fig. 11. We can see that the reconstruction error of CNN-AE can converge to the minimum value with fewest training iteration steps, which indicate that the CNN-AE can extract features which can give the best representing of the original data.

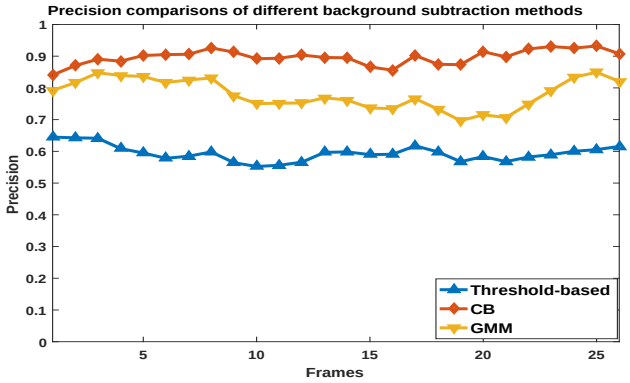

Fig. 8. Precision comparisons of different background subtraction algorithms for different frames in a video sequence

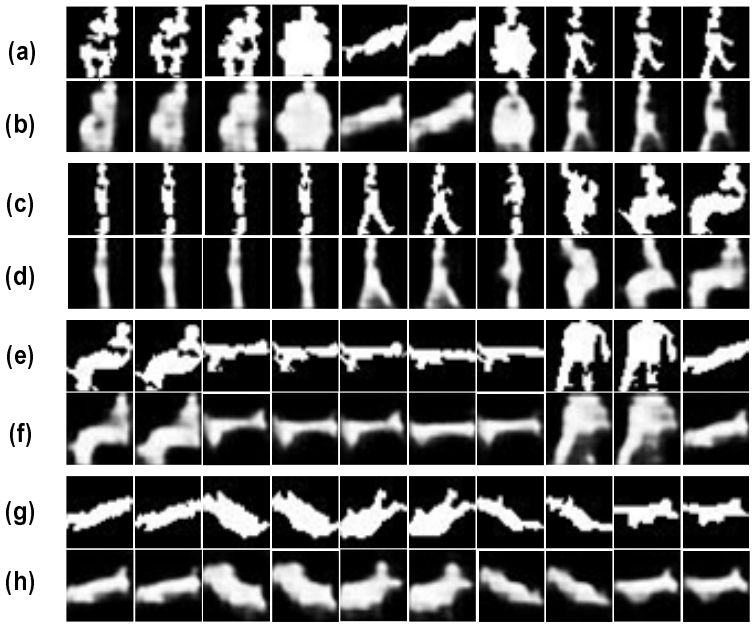

Fig. 9. Original image patches ((a),(c), (e), (g)) and corresponding construction results ((b), (d), (f), (h)) by CNN-AE.

C. Fall Detection System Evaluation

To evaluate the fall detection performance, three evaluation metrics are introduced, which include the true positive rate (TPR), false negative rate (FNR) and total accuracy (TA) defined as:

$$TPR = \frac{No.\ of\ correctly\ detected\ falls}{No.\ of\ correctly\ falls} \tag{15}$$

$$FNR = \frac{No.\ of\ mistaken\ falls}{No.\ of\ normal\ activities} \tag{16}$$

$$TA = \frac{No.\ of\ correctly\ classified\ activities}{No.\ of\ total\ activities} \tag{17}$$

$$\tag{18}$$

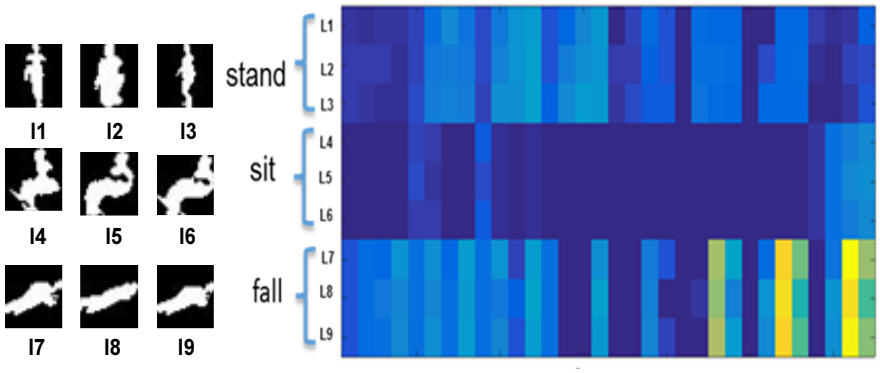

Fig. 10. Silhouettes of different activities (from I1 to I9) and their corresponding visualized extracted features (from L1 to L9). Lines L1-L3 represent visualized feature vectors of silhouettes I1-I3 corresponding to standing; Lines L4-L6 represent visualized feature vectors of silhouettes I4-I6 corresponding to sitting;Lines L7-L9 represent visualized feature vectors of silhouettes I7-I9 corresponding to falling

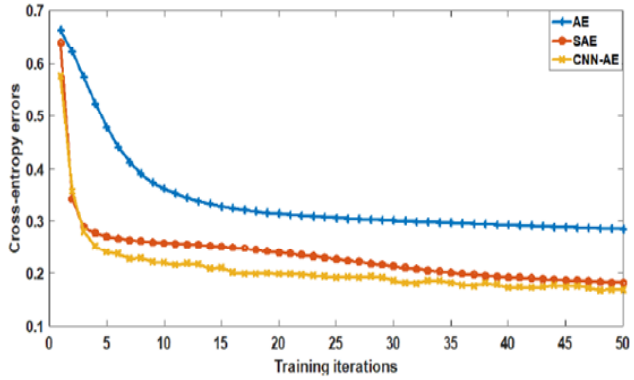

Fig. 11. The comparison between the CNN-AE and other types of autoencoders (AE and SAE) with respect to the cross-entropy reconstruction errors.

For a good fall detection method, TPR should be 1, FNR should be 0 and TA should be 1.

We test a variety of OCSVM models with different kernels for evaluating the fall detection performance on the testing dataset. For each OCSVM model, we have tested its performance by choosing a variety of thresholds in (12). Under each threshold, the TPR, FNR and TA are calculated. The calculated TPRs and FNRs under different thresholds can be plotted as a ROC curve. In Fig. 12 ROC curves for all tested kernels are presented while each ROC curve shows the TPR against the FNR at various threshold settings for a particular kernel. Moreover, the best performance corresponding to the optimal threshold with respect to every kernel is shown in Table, which indicates that compared with other kernels in Table, a Gaussian kernel with $\gamma = 0.5$ can achieve a more accurate fall detection result.

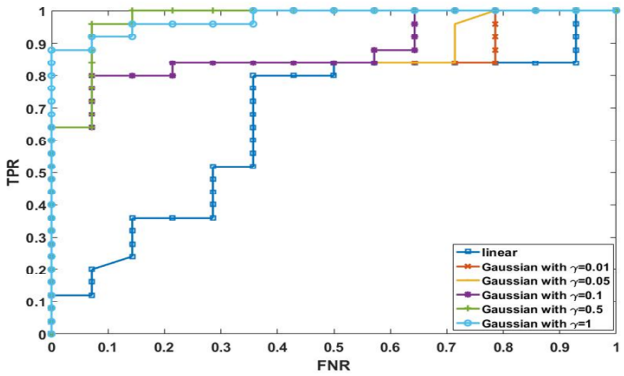

Fig. 12. The ROCs for OCSVM model with different kernels.

TABLE IV
FALL DETECTION PERFORMANCE BY DIFFERENT DIMENTIONALITIES OF CNN-AE FEATURES

		TPR	FNR	Total accuracy
Linear kernel		80%	35.71%	74.36%
Gaussian kernel	$\gamma = 0.01$	80%	7.14%	84.62%
	$\gamma = 0.05$	80%	7.14%	84.62%
	$\gamma = 0.1$	80%	7.14%	84.62%
	$\gamma = 0.5$	100%	8%	**94.87**%
	$\gamma = 1$	96%	14.29%	92.21%

We also investigate the effect of the dimensionality of the extracted CNN-AE features for the fall detection performance. We have modified the network architecture in Table III, to make the number of outputs in layer 6, which are regarded as features the feature dimensionality be different values: 32, 64, 128 and 256. Under each feature dimensionality, we train and test the OCSVM model for fall detection. The results are summarized in Table V. From the table, we can observe that d=64 is most suitable for our application, with higher dimensionality features introducing no extra benefits in improving the total accuracy.

TABLE V
FALL DETECTION PERFORMANCE BY DIFFERENT DIMENTIONALITIES OF CNN-AE FEATURES

	TPR	FNR	Total accuracy
d=32	100%	12%	92.31%
d=64	100%	8%	94.87%
d=128	100%	8%	94.87%
d=256	100%	8%	94.87%

Table VI shows the fall detection performance by exploiting different types of features extracted from human silhouettes in video frames, combined with the OCSVM model for fall detection. Related features include: i). 30*30 original image patches ii). combination of aspect ratios and orientation angles as in [39] iii). projection histogram [40] iv). features obtained by the principle component analysis (PCA) v). features extracted from the CNN-AE as presented in our work. From the related results, we can see that by the aid of CNN-AE, more representative features can be extracted to obtain better performance for distinguishing falls and non-falls.

TABLE VI
FALL DETECTION PERFORMANCE BY DIFFERENT TYPES OF FEATURES

	TPR	FNR	Total accuracy
original image patches	85.71%	4%	92.31%
aspect ratio+angle	100%	16%	89.74%
projection histogram	100%	28%	82.05%
PCA	78.57%	12%	84.62%
CNN-AE	100%	8%	**94.87**%

TABLE VII
COMPARISONS OF DIFFERENT MODELS FOR FALL DETECTION

	TPR	FNR	Total accuracy
GMM model	96%	14.29%	92.31%
Two-class model	100%	8%	94.87%
OCSVM model (proposed)	100%	8%	94.87%

Finally, we compared the proposed OCSVM model in the paper with two-class model using both the normal activities data and fall data for fall detection (here the two-class model in this work is implemented as a two-class SVM model, which is widely adopted for both fall detection and other applications) as well as another Gaussian mixture model which also relying on only the normal activities data for fall detection as in [27]–[29]. The results are summarized in Table VII. For fair comparison, the same CNN-AE features are used with these three models for fall detection. We can see that the proposed OCSVM model achieves better performance than the GMM, with high TPR and accuracy with lower FNR value. Although the two-class model archives the same performance as the proposed approach, however, more data (both the normal activities data and fall data) is needed for training the two-class model to detect fall. By the aid of the OCSVM model, good fall detection performance can be guaranteed with less data being required.

IV. CONCLUSION

In this paper, we proposed a computer vision based data driven modeling approach for achieving person specific fall detection. Video recordings of an elderly person's daily activities are made by a single camera mounted at home, which covers the living area of a resident. Human silhouettes from video recordings are extracted, with related low-dimensionality features representing silhouettes being extracted by the CNN-AE. A OCSVM model is then trained based on extracted features from recorded video data. And this data driven OCSVM model can be applied for determining whether falling happens or not based on CNN-AE features extracted in the testing video recordings. The experimental results show that proposed system can effectively detect falls in a real home environment with a low false alarm rate. The proposed system constructs a data driven model based on the video recording of a particular monitored resident's daily activities himself/herself for fall detection. In this way, the built model is person-specific, which is more suitable to detect falls based on the resident's own living environment and activity characteristics; besides, the model can be constructed automatically from a collection of video frames with the minimal human intervenes (no need for manually segmentation/annotation), which makes its application at home more convenient. The proposed approach only relies on normal activities data without needing the falling data, thus reducing the data requirement for machine learning model construction for fall detection.

In future works, we will take the changes of the daily activities of residents into account. Instead of adopting a fixed model, strategies will be developed to update both the CNN autoencoder for feature extractions and the OCSVM model for the data description in an adaptive way, to accommodate for the resident's daily activities changes across a comparatively longer time period for targeting at a life-long activity monitoring system; besides, not only limited to falls, we will target at detecting more kinds of abnormal motions (such as Parkinson gait).

REFERENCES

[1] Z. Pang, L. Zheng, J. Tain, S. Kao-Walter, E. Dubrova, and Q. Chen, "Design of a terminal solution for integration of in-home health care devices and services towards the internet-of-things," *Enterprise Information Systems*, vol. 9, no. 1, pp. 86–116, 2015.

[2] "Families and households in the uk: 2016," *Available: https://www.ons.gov.uk/peoplepopulationandcommunity*, 2016.

[3] "Falls and fracture consensus statement supporting commissioning for prevention," *Available: https://www.england.nhs.uk/south/wp-content/uploads/sites/6/2017/03/falls-fracture.pdf*, 2017.

[4] "Web based injury statistics query and reporting system," *Available: https://webappa.cdc.gov/sasweb/ncipc/nfilead.html*, 2016.

[5] T. Al-Aama, "Falls in the elderly: Spectrum and prevention," *Can Fam Physician*, vol. 57, no. 7, pp. 771–776, 2011.

[6] "Falls," *Available: https://www.who.int/news-room/fact-sheets/detail/falls*, 2018.

[7] S. T. L. K. R. Broadley, J. Klenk and M. Granat, "Methods for the real-world evaluation of fall detection technology: A scoping review," *Sensors*, vol. 18, no. 7, pp. 1–28, 2018.

[8] W. Cheng and D. Jhan, "Triaxial accelerometer-based fall detection method using a self-constructing cascade-adaboost-svm classifier," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 2, pp. 411–419, 2013.

[9] L. Tong, Q. Song, Y. Ge, and M. Liu, "Hmm-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1849–1859, 2013.

[10] M. Cheffena, "Fall detection using smartphone audio features," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 4, pp. 1073–1080, 2016.

[11] Y. Li, K. Ho, and M. Popescu, "Efficient source separation algorithms for acoustic fall detection using a microsoft kinect," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 745–755, 2014.

[12] M. Khan, M. Yu, P. Feng, L. Wang, and J. Chambers, "An unsupervised acoustic fall detection system using source separation for sound interference suppression," *Signal Processing*, vol. 110, pp. 199–210, 2015.

[13] A. Shahzad and K. Kim, "Falldroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 35–44, 2018.

[14] H. Kerdegari, S. Mokaram, S. Mokaram, K. Samsudin, K. Samsudin, and A. Ramli, "A pervasive neural network based fall detection system on smart phone," *Journal of Ambient Intelligence and Smart Environments*, vol. 7, no. 2, 2016.

[15] M. Alwan, P. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," *2nd International Conference on Information and Communication Technologies, Damascus, Syria*, 2006.

[16] B. Su, K. Ho, M. R, and S. Marjorie, "Radar placement for fall detection: Signature and performance," *Journal of Ambient Intelligence and Smart Environments*, vol. 10, no. 1, pp. 21–34, 2018.

[17] L. Liang, P. Mihail, S. Marjorie, R. Marilyn, and C. Paul, "An automatic in-home fall detection system using doppler radar signatures," *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 4, pp. 453–466, 2016.

[18] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution," *IEEE Transactions on Information Technology in Biomedicine,*, vol. 15, no. 2, pp. 290–300, 2011.

[19] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "3d head tracking for fall detection using a single calibrated camera," *Image and Vision Computing*, vol. 31, no. 3, pp. 246–254, 2013.

[20] K. Ozcan, S. Velipasalar, and P. Varshney, "Aotonomous fall detection with wearable cameras by using relative entropy distance measure," *IEEE Transactions on Human Machine Systems*, vol. 47, no. 1, pp. 31–39, 2017.

[21] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1522–1532, 2008.

[22] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "Automatic monocular system for human fall detection based on variations in silhouette area," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 427–436, 2013.

[23] E. Stone and M. Skubic, "Fall detection in home of older adults using the microsoft kinect," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 290–301, 2015.

[24] M. Yu, L. Gong, and S. Kollias, "Computer vision based fall detection by a convolutional neural network," *19th ACM International Conference on Multimodal Interaction, Glasgow, UK*, 2017.

[25] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional cnn combined with lstm on video kinematic data," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 1, pp. 314–323, 2019.

[26] D. Glen, M. Marc, D. Mieke, V. Ellen, D. Els, D. Eddy, M. Koen, T. Jos, C. Tom, G. Toon, T. Tinne, and V. Bart, "Camera-based fall detection using real-world versus simulated data: how far are we from the solution?" *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 2, pp. 149–168, 2016.

[27] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 2011.

[28] X. Zhuang, J. Huang, G. Potamianos, and M. Hasegawa-Johnson, "Acoustic fall detection using gaussian mixture models and gmm supervectors," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.

[29] A. Poonsri and W. Chiracharit, "Fall detection using gaussian mixture model and principle component analysis," *9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2017.

[30] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[31] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy*, 2017.

[32] R. Gonzalez, "Digital image processing," *Third Edition, Prentice Hall*, 2008.

[33] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *International Conference on Artificial Neural Networks,Espoo, Finland*, 2011.

[34] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *The MIT Press*, 2016.

[35] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 11, pp. 2298–3304, 2017.

[36] C. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.

[37] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.

[38] "Keras: The python deep learning library," *Available: https://keras.io/*, 2018.

[39] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), Niagara Falls, Ont., Canada*, 2007.

[40] M. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE transactions on information technology in biomedicine*, vol. 16, no. 6, pp. 1274–1286, 2012.